Lógica

Estrategias de Resolución

Damiano Zanardini

Graduado/a en Ingeniería Informática
Facultad de Informática
Universidad Politécnica de Madrid
damiano@fi.upm.es

Curso Académico 2010/2011

Introducción

El problema

- el método de *saturación*, tal y como está escrito, genera muchas cláusulas irrelevantes o redundantes
- es necesario usar unas *reglas de selección* sistemáticas que simplifiquen el proceso y lo hagan *computacionalmente eficiente*
- dos tipos de criterios
  - *estrategias de simplificación*: reducir el número de cláusulas
  - *estrategias de refinamiento*: limitar la generación de cláusulas

Terminología

- $\mathcal{C}$ es el conjunto inicial de cláusulas
- $\mathcal{C}'$ es el conjunto de cláusulas *actual* (en algún momento durante el procedimento de deducción cuando queremos aplicar una regla)

Estrategias de Simplificación

1 Eliminación de cláusulas idénticas

- es trivial que $\mathcal{C} \vdash_{UMG} \square$ sii se puede derivar $\square$ eliminando las cláusulas idénticas (a parte una copia)

Cómo hacerlo

- si se genera una cláusula que aparece ya en $\mathcal{C}'$ esta cláusula no se añade

☞ se note que en un programa de ordenador esto puede suponer que se deba comparar la nueva cláusula con todas las demás en $\mathcal{C}'$

Estrategias de Simplificación

2 Eliminación de cláusulas con literales puros

- un literal L se dice *puro* sii no existe en el conjunto otro literal $\neg L'$ tal que L y L' son unificables
- $\mathcal{C} \vdash_{UMG} \square$ sii se puede derivar $\square$ después de eliminar de $\mathcal{C}$ las cláusulas con literales puros
  - una cláusula con literales puros es inutil de cara a la refutación porque nunca se podrá eliminarla por resolución

Como hacerlo

- las cláusulas con literales puros se eliminan del conjunto
- es suficiente aplicar esta estrategia *una vez*, porque no se generarán nuevas cláusulas con literales puros

Estrategias de Simplificación

3 Eliminación de cláusulas tautológicas

- $\mathcal{C} \vdash_{UMG} \square$ sii se puede derivar $\square$ a partir de $\mathcal{C}$ después de eliminar las tautologías

Como hacerlo

- si se genera una cláusula tautológica no se añade a $\mathcal{C}'$

Estrategias de Simplificación

Ejemplo: $\mathcal{C} = \{p \vee q,\ \neg p \vee q,\ p \vee \neg q,\ \neg p \vee \neg q\}$

- aplicando todas las reglas de simplificación la derivación queda

(1)	$p \vee q$	
(2)	$\neg p \vee q$	
(3)	$p \vee \neg q$	
(4)	$\neg p \vee \neg q$	
(5)	q	(1,2)
(6)	p	(1,3)
(7)	$\neg p$	(2,4)
(8)	$\neg q$	(3,4)
(9)	$\square$	(5,8)

- hay que notar que no se ha usado ninguna otra estrategia *inteligente*

	pares de cláusulas	resolventes generados
primera iteración	6 (6)	8 (8)
segunda iteración	19 (60)	9 (....)

Derivaciones y Refutaciones

Derivaciones

Una *derivación* de C a partir de $\{C_1, .., C_n\}$ es una secuencia $\langle C_1, .., C_n, R_1, .., R_m \rangle$ tal que

- cada R_i es el resolvente de dos cláusulas anteriores
- no se realiza el mismo paso de resolución más de una vez
- $R_m = C$

Refutaciones

Una *refutación* de $\{C_1, .., C_n\}$ es una derivación de $\square$ a partir de $\{C_1, .., C_n\}$

Hechos

- una derivación es una deducción correcta (corrección de la resolución)
- si $INSAT(\mathcal{C})$ entonces hay una refutación de $\mathcal{C}$ (completitud de la resolución)

Resolución Lineal

Derivaciones lineales

Una derivación lineal de C_m a partir de $\{C_1, .., C_n\}$ es una secuencia

$$\langle C_1, .., C_n, C_{n+1}, .., C_m \rangle$$

tal que

- C_{n+1} es el resolvente de dos cláusulas de $\{C_1, .., C_n\}$ (*cláusulas de cabezera*)
- para todo $i > n+1$, C_i es el resolvente de C_{i-1} y otra cláusula C_j, con $j < i-1$

Resolución Lineal

Propiedades

La resolución lineal es *completa*: $INSAT(\mathcal{C})$ sii existe una refutación lineal de $\mathcal{C}$

- las derivaciones se pueden restringir a derivaciones lineales
- los árboles de búsqueda se pueden restringir a árboles lineales

☞ ¿qué pasa con el árbol de búsqueda y el árbol de resolución anteriores?

En una derivación de C a partir de $\mathcal{C}$, no es necesario probar con todas las cláusulas en $\mathcal{C}$ como punto de partida de la refutación (de $\neg C$)

- si un conjunto $\mathcal{C}$ es satisfacible y $\mathcal{C} \cup \neg C$ no lo es, entonces existe una refutación lineal que empieza con $\neg C$

Resolución Input

Derivaciones input

Una derivación input de C_m a partir de $\{C_1, .., C_n\}$ es una secuencia

$$\langle C_1, .., C_n, C_{n+1}, .., C_m \rangle$$

tal que

- para todo $i > n$ C_i es un resolvente de $C_k \in \{C_1, .., C_n\}$ y otra C_j $(j < i)$

Ejemplo

$$C_1 = \neg p(x) \vee q(x) \qquad C_2 = \neg r(x) \vee \neg q(x) \qquad C_3 = r(a)$$
$$C_4 = s(a), \qquad C_5 = \neg s(x) \vee p(x)$$

- refutación input a partir de C_1:

$$\begin{array}{ll} R_1 = \neg p(x) \vee \neg r(x) & (C_1, C_2) \\ R_2 = \neg s(x) \vee \neg r(x) & (R_1, C_5) \\ R_3 = \neg s(a) & (R_2, C_3) \\ R_4 = \square & (R_3, C_4) \end{array}$$

Resolución Input

Derivaciones input

Una derivación input de C_m a partir de $\{C_1, .., C_n\}$ es una secuencia

$$\langle C_1, .., C_n, C_{n+1}, .., C_m \rangle$$

tal que

- para todo $i > n$ C_i es un resolvente de $C_k \in \{C_1, .., C_n\}$ y otra C_j $(j < i)$

Ejemplo

$$C_1 = \neg p(x) \vee q(x) \qquad C_2 = \neg r(x) \vee \neg q(x) \qquad C_3 = r(a)$$
$$C_4 = s(a), \qquad C_5 = \neg s(x) \vee p(x)$$

- refutación input a partir de C_5:

$$\begin{array}{ll} R_1 = p(a) & (C_4, C_5) \\ R_2 = q(a) & (R_1, C_1) \\ R_3 = \neg r(a) & (R_2, C_2) \\ R_4 = \square & (R_3, C_3) \end{array}$$

Resolución Input

Ejemplo: $C_1 = p \vee q,\ C_2 = \neg q,\ C_3 = r \vee q,\ C_4 = \neg r$

- refutación *input no lineal* a partir de C_1:
  $R_1 = p \quad (C_1, C_2)$
  $R_2 = r \quad (C_2, C_3)$
  $R_3 = \square \quad (R_2, C_4)$
- dado que R_1 no interviene en el resto de la derivación se puede construir una refutación *input lineal* a partir de la anterior:
  $R_1 = r \quad (C_2, C_3)$
  $R_2 = \square \quad (R_1, C_4)$

Resolución Input

Lema

Dada una derivación input no lineal de R_m se puede construir una derivación input lineal de R_m

Demostración.

Sea $C_1, .., C_n, R_1, .., R_m$ una derivación input no lineal de R_m

1. sea R_{k+1} $(n+1 \leq k \leq m)$ el primer resolvente que no se ha generado linealmente
2. R_{k+1} es el resolvente de $C \in \{C_1, .., C_n\}$ y R_j $(1 \leq j < k)$
3. por resolución input, R_{k+1} y R_k no se pueden resolver entre sí
4. por ❸ se pueden generar dos derivaciones independientes
   - $C_1, .., C_n, R_1, .., R_k, ..$ (lineal hasta R_k)
   - $C_1, .., C_n, R_1, .., R_j, R_{k+1}, ..$ (lineal hasta R_{k+1})
5. una de estas derivaciones terminará en R_m

☞ se puede *linearizar* dicha derivación aplicándole más veces este lema

Resolución Input

(contra)Ej.: $C_1 = p \vee q$, $C_2 = \neg p \vee q$, $C_3 = r \vee \neg q$, $C_4 = \neg r \vee \neg q$

- *no input y no lineal*:
  $R_1 = q \vee q \quad (C_1, C_2)$
  $R_2 = \neg q \vee \neg q \quad (C_3, C_4)$
  $R_3 = \square \quad (R_1, R_2)$
- para toda derivación no lineal hay una lineal equivalente:
  $R_1 = q \vee q \quad (C_1, C_2)$
  $R_2 = r \quad (R_1, C_3)$
  $R_3 = \neg q \quad (R_2, C_4)$
  $R_4 = \square \quad (R_3, R_1)$
- ¿se puede encontrar una derivación input para toda derivación no input?

Resolución Input

La resolución input *no* es completa

No se puede decir que para cada conjunto insatisfacible de cláusulas existe una refutación input

Ej. $p \vee q$
$\neg p \vee r$
$p \vee \neg q$
$s \vee q$
$s \vee \neg q$
$\neg s \vee \neg r$

Resolución Dirigida (Wos-Robinson-Carson, 1965)

Derivaciones dirigidas

Una derivación dirigida de C_m a partir de $\{C_1, .., C_n\}$, con conjunto soporte $S \subset \mathcal{C}$, es una secuencia $\langle C_1, .., C_n, C_{n+1}, .., C_m \rangle$ tal que

- para todo $i > n$, C_i es resolvente de dos cláusulas anteriores en la secuencia que no pertenecen ambas a S
- las cláusulas de S son *cláusulas de soporte*, y las de $\mathcal{C} \setminus S$ son *cláusulas objetivo*
- esta técnica se motiva con el hecho que:
  - supongamos que se quiera demostrar B a partir de $A_1 \wedge .. \wedge A_k$
  - es decir, que $A_1 \wedge .. \wedge A_k \wedge \neg B$ es insatisfacible
  - en este caso, $A_1 \wedge .. \wedge A_k$ es normalmente satisfacible en sí
  - por lo tanto puede ser útil evitar de resolver dos cláusulas de este conjunto entre sí
  - el conjunto soporte identifica el subconjunto de $\mathcal{C}$ que se supone satisfacible (el resultado a demostrar no pertenece al conjunto soporte)

Resolución Dirigida (Wos-Robinson-Carson, 1965)

Ejemplo

$$\begin{aligned}\mathcal{C} &= \{C_1 = s \vee t,\quad C_2 = \neg s \vee p,\quad C_3 = \neg q \vee r,\quad C_4 = q \vee \neg p,\\ &\qquad C_5 = u \vee \neg r,\quad C_6 = \neg u,\quad C_7 = \neg t\}\\ S &= \{C_1,\ C_2,\ C_3,\ C_4,\ C_5\}\end{aligned}$$

dirigida	
$R_1 = s$	(C_1, C_7)
$R_2 = p$	(R_1, C_2)
$R_3 = q$	(R_2, C_4)
$R_4 = r$	(R_3, C_3)
$R_5 = u$	(R_4, C_5)
$R_6 = \square$	(R_5, C_6)

no dirigida	
$R_1 = t \vee p$	(C_1, C_2)
$R_2 = p$	(R_1, C_7)
$R_3 = q$	(R_2, C_4)
$R_4 = r$	(R_3, C_3)
$R_5 = u$	(R_4, C_5)
$R_6 = \square$	(R_5, C_6)

Resolución Dirigida (Wos-Robinson-Carson, 1965)

Propiedades

La resolución dirigida es *completa*: si $INSAT(\mathcal{C})$ y $S \subset \mathcal{C}$ es satisfacible entonces existe una refutación dirigida de $\mathcal{C}$ con conjunto soporte S

- esto no es muy útil si no se da un método para hallar un S satisfacible

Heurística para encontrar a S

En la práctica, cuando se busca una refutación de una conclusión a partir de unas premisas, es razonable suponer que las premisas sean satisfacibles

- premisas: S
- negación de la conclusión (forma clausular): $\mathcal{C} \setminus S$
- si las premisas son inconstistentes entonces cualquier conclusión se puede derivar
- pero, si son consistentes, entonces $\square$ se puede derivar de la conclusión negada

Resolución Ordenada

Derivaciones ordenadas

Una derivación ordenada de C_m a partir de $\{C_1, .., C_n\}$ es una secuencia

$$\langle C_1, .., C_n, C_{n+1}, .., C_m \rangle$$

tal que

- para todo $i > n$, C_i es el resolvente de dos cláusulas anteriores

  $$A_1 \vee L_{11} \vee .. \vee L_{1p} \qquad \text{y} \qquad \neg A_2 \vee L_{21} \vee .. \vee L_{2q}$$

  donde A_1 y A_2 son unificables con $UMG\ \sigma$ y se mantiene el orden
- los literales de C_i están ordenados de esta forma:

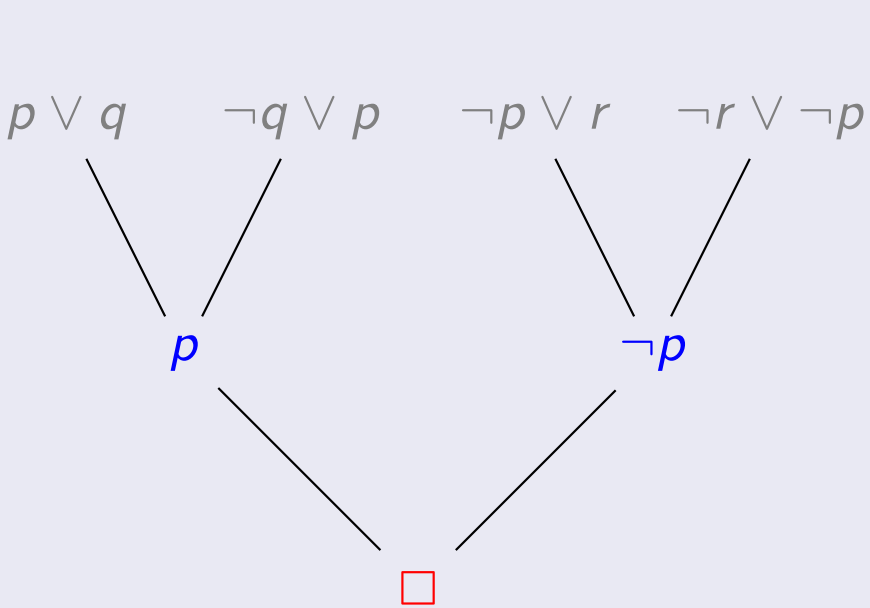
  $$(L_{11} \vee .. \vee L_{1p} \vee L_{21} \vee .. \vee L_{2q})\sigma$$

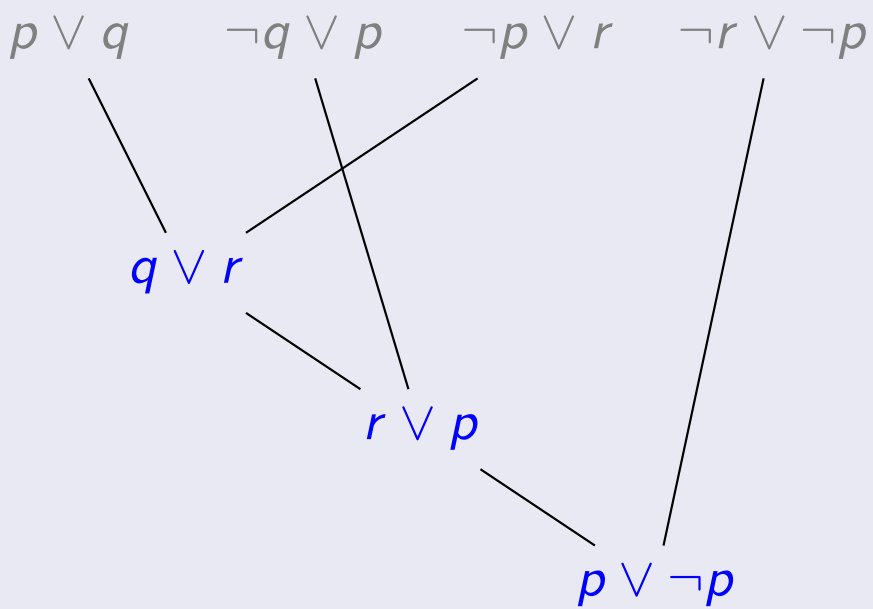
Resolución Ordenada

(No) Propiedades

La resolución ordenada *no* es completa

contraejemplo: $\{p \vee q, \quad \neg q \vee p, \quad \neg p \vee r, \quad \neg r \vee \neg p\}$





Resumen

Corrección y completitud

- Corrección: se puede derivar $\square$ sólo si $INSAT(\mathcal{C})$
- Completitud: si $INSAT(\mathcal{C})$ entonces se puede derivar $\square$

	correcta	completa
lineal	✓	✓
input	✓	no
dirigida	✓	✓(si $SAT(S)$)
ordenada	✓	no